

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (original) A method in a data processing system having at least one host and host memory and a system interface coupling said host and said host memory to a network interface device, comprising:
 - a. receiving an IP packet from a network interface;
 - b. performing TCP/IP processing on said IP packet in said network interface device, resulting in TCP data from said IP packet;
 - c. determining if SSL processing is required;
 - d. performing SSL processing on said TCP data that require SSL processing, resulting in data from said TCP data; and
 - e. transmitting said data to said system interface.
2. (original) The method of claim 1 further comprising a TCP/IP processor or TCP/IP processors performing TCP/IP processing on said IP packet.
3. (original) A method as in claim 1 wherein said host performs TCP/IP processing on said IP packet while TCP payload data remains in said network interface device.
4. (original) The method of claim 1 further comprising a buffer or buffers which may be utilized for internal processing.

5. (original) The method of claim 1 further comprising examining the TCP connection states of said TCP data to determine if SSL processing is required.
6. (original) The method of claim 1 further comprising a cryptographic acceleration device for performing SSL processing on said TCP data, wherein said SSL processing does not utilize system memory, the system bus or the chip interconnection network.
7. (original) The method of claim 6 further comprising transmitting only packets that require SSL processing to said cryptographic acceleration device and transmitting packets that do not require SSL processing to said system interface.
8. (original) The method of claim 6 further comprising accessing an SSL database necessary for SSL processing directly from said cryptographic acceleration device, wherein said SSL database may exist in hardware, or said SSL database may exist in software.
9. (original) A method as in claim 6 wherein said cryptographic acceleration device is an SSL Cryptographic Accelerator.
10. (original) The method of claim 1 further comprising transmitting said IP packet to a storage location located in said network interface device from said network interface, wherein said TCP/IP processing is performed in said storage location, resulting in said TCP data from said IP packet.

11. (original) A method as in claim 10 wherein said transmitting said IP packet to said storage location from said network interface comprises transmitting said IP packet by Direct Memory Access (DMA) operation.
12. (original) A method as in claim 10 wherein said storage location comprises Network Offload Memory (NOM).
13. (original) A method as in claim 10 wherein said storage location comprises a temporary buffer.
14. (original) The method of claim 10 further comprising a cryptographic acceleration device for performing SSL processing on said TCP data, wherein said SSL processing does not utilize system memory, the system bus or the chip interconnection network.
15. (original) The method of claim 14 further comprising a DMA engine or DMA engines sending only packets that require SSL processing to said cryptographic acceleration device and sending packets that do not require SSL processing to said system interface.
16. (original) The method of claim 14 further comprising transmitting said data from said cryptographic acceleration device to said system interface by Direct Memory Access (DMA) operation

17. (original) A method in a data processing system having at least one host and host memory and a system interface coupling said host and said host memory to a network interface device, comprising:
- a. receiving data from said system interface;
 - b. determining if SSL processing is required;
 - c. performing SSL processing on said data that require SSL processing, creating an SSL packet;
 - d. performing TCP/IP processing on said SSL packet in said network interface device, creating an IP packet; and
 - e. transmitting said IP packet to a network interface.
18. (original) The method of claim 17 further comprising a TCP/IP processor or TCP/IP processors performing TCP/IP processing on said SSL packet, and creating an IP packet.
19. (original) A method as in claim 17 wherein said host performs TCP/IP processing on said SSL packet while said SSL packet payload remains in said network interface device.
20. (original) The method of claim 17 further comprising examining the TCP connection states of said data to determine if SSL processing is required.
21. (original) The method of claim 17 further comprising examining an indication from the host to determine if SSL processing is required.

22. (original) The method of claim 17 further comprising a cryptographic acceleration device for performing SSL processing on said data, wherein said SSL processing does not utilize system memory, the system bus or the chip interconnection network.
23. (original) The method of claim 22 further comprising accessing an SSL database necessary for SSL processing directly from said cryptographic acceleration device, wherein said SSL database may exist in hardware, or said SSL database may exist in software.
24. (original) A method as in claim 22 wherein said cryptographic acceleration device is an SSL Cryptographic Accelerator.
25. (original) The method of claim 17 further comprising a buffer or buffers which may be utilized for internal processing.
26. (original) The method of claim 17 further comprising transmitting said SSL packet to a storage location located in said network interface device, wherein said TCP/IP processing is performed in said storage location, creating an IP packet.
27. (original) A method as in claim 26 wherein said storage location comprises a Network Offload Memory (NOM).
28. (original) A method as in claim 26 wherein said storage location comprises a temporary buffer.

29. (original) The method of claim 26 further comprising a cryptographic acceleration device for performing SSL processing on said data, wherein said SSL processing does not utilize system memory, the system bus or the chip interconnection network.
30. (original) The method of claim 29 further comprising transmitting said data from said system interface to said cryptographic acceleration device by Direct Memory Access (DMA) operation.
31. (original) The method of claim 29 further comprising a DMA engine or DMA engines sending only packets that require SSL processing to said cryptographic acceleration device and sending packets that do not require SSL processing to said storage location.
32. (original) The method of claim 26 further comprising transmitting said IP packet from said storage location to said network interface by Direct Memory Access (DMA) operation.
33. (original) An apparatus, comprising:
- a. a network interface or network interfaces, said network interface or network interfaces receive and send IP packets;
 - b. a TCP/IP processor or TCP/IP processors coupled to said network interface or network interfaces, said TCP/IP processor or TCP/IP processors perform TCP/IP processing;

- c. an accelerator or accelerators coupled to said network interface or network interfaces, said accelerator or accelerators perform SSL processing on inbound data, and/or perform SSL processing on outbound data; and
- d. a system interface or system interfaces coupled to said accelerator or accelerators, said system interface or system interfaces receive and/or send data from and/or to a system.

34. (original) The apparatus of claim 33 further comprising a single device or multiple devices.

35. (original) An apparatus as in claim 33 wherein said apparatus comprises:

- a. an inbound network interface, said inbound network interface receives inbound IP packets from a network;
- b. a chip interconnection network coupled to said inbound network interface;
- c. a TCP/IP processor coupled to said chip interconnection network, said TCP/IP processor performs TCP/IP processing on said IP packet;
- d. an accelerator coupled to said chip interconnection network, said accelerator performs SSL processing;
- e. an SSL database coupled to said accelerator; and
- f. a system interface coupled to said accelerator, said system interface sends data to a system.

36. (original) An apparatus as in claim 35 wherein said inbound network interface comprises an Ethernet interface.

37. (original) The apparatus of claim 35 wherein said accelerator is an SSL Cryptographic Accelerator, said SSL Cryptographic Accelerator does not utilize system memory, system bus or chip interconnection network.
38. (original) The apparatus of claim 35 further comprising a storage location coupled to said chip interconnection network, wherein said TCP/IP processor accesses the IP packet at the storage location and performs the TCP/IP processing on said IP packet.
39. (original) An apparatus as in claim 38 wherein said storage location comprises a Network Offload Memory (NOM).
40. (original) An apparatus as in claim 38 wherein said storage location comprises a temporary buffer.
41. (original) The apparatus of claim 38 wherein said IP packets received from said inbound network interface are transmitted to said storage location by Direct Memory Access (DMA) operation.
42. (original) The apparatus of claim 38 wherein said SSL packets in said storage location are transmitted to said accelerator by Direct Memory Access (DMA) operation.
43. (original) The apparatus of claim 38 wherein said data in said storage location are transmitted to said system interface by Direct Memory Access (DMA) operation.

44. (original) An apparatus as in claim 33 wherein said apparatus comprises:

- a. a system interface, said system interface receives data from a system.
- b. an accelerator coupled to said system interface, said accelerator performs SSL processing on said data, creating an SSL packet;
- c. an SSL database coupled to said accelerator;
- d. a chip interconnection network coupled to said accelerator;
- e. a TCP/IP processor coupled to said chip interconnection network, said TCP/IP processor performs TCP/IP processing on the SSL packet and creating an IP packet; and
- f. a outbound network interface coupled to said chip interconnection network, said outbound network interface sends outbound IP packets to a network;

45. (original) An apparatus as in claim 44 wherein said outbound network interface comprises an Ethernet interface.

46. (original) The apparatus of claim 44 wherein said accelerator is an SSL Cryptographic Accelerator, wherein said SSL Cryptographic Accelerator does not utilize system memory, system bus or chip interconnection network.

47. (original) The apparatus of claim 44 further comprising a storage location coupled to said chip interconnection network, wherein said TCP/IP processor performs TCP/IP processing at the storage location.

48. (original) The apparatus of claim 47 wherein said storage location comprises a Network Offload Memory (NOM).

49. (original) The apparatus of claim 47 wherein said storage location comprises a temporary buffer.
50. (original) The apparatus of claim 47 wherein said data received from said system interface are transmitted to said storage location by Direct Memory Access (DMA) operation.
51. (original) The apparatus of claim 47 wherein said SSL packet in said accelerator are transmitted to said storage location by Direct Memory Access (DMA) operation.
52. (original) The apparatus of claim 47 wherein said IP packet at the storage location are transmitted to said outbound network interface by Direct Memory Access (DMA) operation.
53. (original) An apparatus as in claim 33 wherein said accelerator or accelerators further comprise a buffer or buffers.
54. (original) An apparatus as in claim 35 wherein said accelerator further comprises a buffer.
55. (original) An apparatus as in claim 35 wherein said SSL database may exist in software, or said SSL database may exist in hardware.

56. (original) An apparatus as in claim 44 wherein said accelerator further comprises a buffer.

57. (original) An apparatus as in claim 44 wherein said SSL database may exist in software, or said SSL database may exist in hardware.

58. (new) A computing system, comprising:

- a central processing unit;

- a host memory coupled to said central processing unit, said host memory to store instructions executed by said central processing unit and data operated upon by way of said central processing unit executing said instructions;

- a network interface to receive inbound IP packets and send outbound IP packets;

- an offload processing subsystem, said offload processing subsystem communicatively coupled to said central processing unit and said network interface, said offload processing subsystem comprising:

- a) TCP/IP logic circuitry for processing TCP/IP tasks on said inbound and outbound IP packets without using said central processing unit and said host memory;

- b) SSL decryption logic circuitry for processing SSL tasks on said inbound IP packets without using said central processing unit and said host memory;

- c) SSL encryption logic circuitry for processing SSL tasks on said outbound IP packets without using said central processing unit and said host memory;

- d) a storage resource comprising one or more memory chips coupled to both said SSL decryption logic circuitry and said SSL encryption logic circuitry, said storage resource to store SSL processing information;
- e) an offload memory coupled to said TCP/IP logic circuitry, said offload memory to store said inbound and outbound TCP/IP packets, said network interface coupled to said offload memory;
- f) a first direct memory access (DMA) controller to retrieve said inbound IP packets from said offload memory, said first DMA controller coupled to said SSL decryption logic circuitry;
- g) a second direct memory access (DMA) controller to retrieve said outbound IP packets from said host memory, said second DMA controller coupled to said SSL encryption logic circuitry;

59. (new) An computing system as in claim 58 wherein said inbound network interface comprises an Ethernet interface.

60. (new) The computing system as in claim 58 comprising a first buffer downstream from an output of said first DMA controller and upstream from an input of said SSL decryption logic circuitry, and, a second buffer downstream from an output of said second DMA controller and upstream from an input of said SSL encryption logic circuitry.